# 8.7: Appendix - Felsenstein's Pruning Algorithm

Felsenstein's pruning algorithm (1973) is an example of dynamic programming, a type of algorithm that has many applications in comparative biology. In dynamic programming, we break down a complex problem into a series of simpler steps that have a nested structure. This allows us to reuse computations in an efficient way and speeds up the time required to make calculations.

The best way to illustrate Felsenstein's algorithm is through an example, which is presented in the panels below. We are trying to calculate the likelihood for a three-state character on a phylogenetic tree that includes six species.

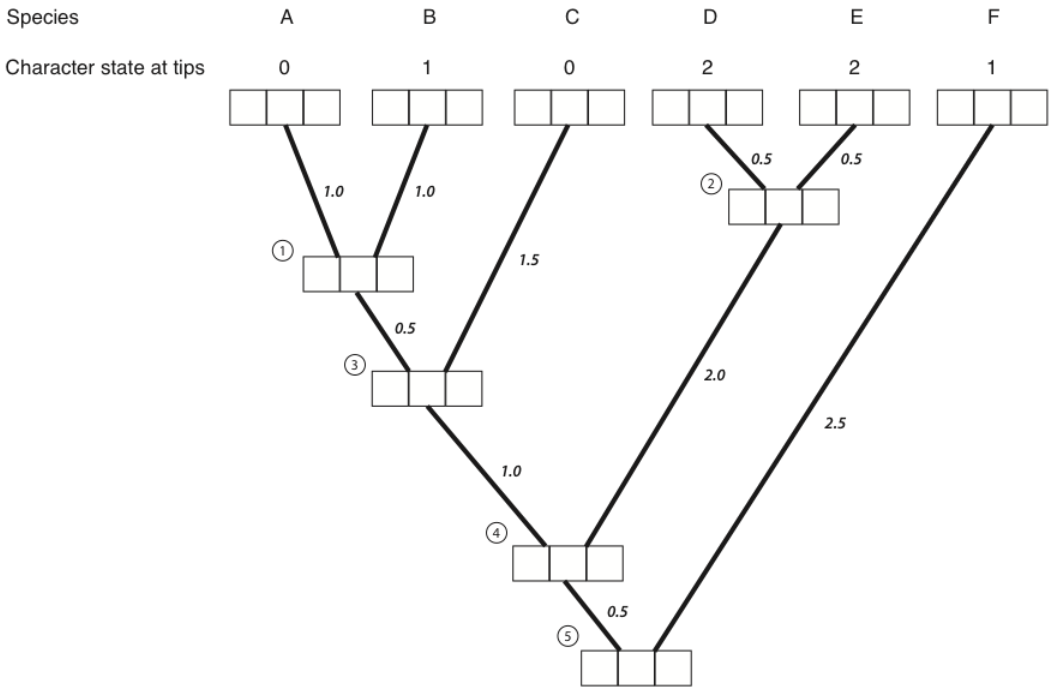


Figure 8.4A. Each tip and internal node in the tree has three boxes, which will contain the probabilities for the three character states at that point in the tree. The first box represents a state of 0, the second state 1, and the third state 2. Image by the author, can be reused under a CC-BY-4.0 license.

1. The first step in the algorithm is to fill in the probabilities for the tips. In this case, we know the states at the tips of the tree. Mathematically, we state that we know precisely the character states at the tips; the probability that that species has the state that we observe is 1, and all other states have probability zero:

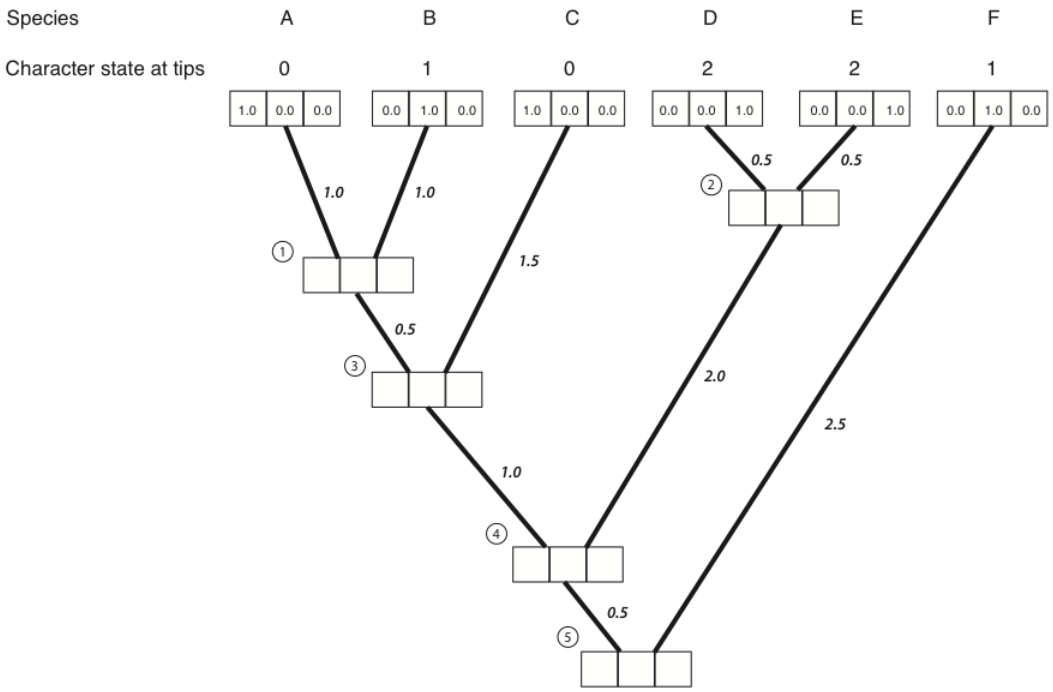


Figure 8.4B. We put a one in the box that corresponds to the actual character state and zeros in all others. Image by the author, can be reused under a CC-BY-4.0 license.

2. Next, we identify a node where all of its immediate descendants are tips. There will always be at least one such node; often, there will be more than one, in which case we will arbitrarily choose one. For this example, we will choose the node that is the most recent common ancestor of species A and B, labeled as node 1 in Figure 8.2B.

3. We then use equation 7.6 to calculate the conditional likelihood for each character state for the subtree that includes the node we chose in step 2 and its tip descendants. For each character state, the conditional likelihood is the probability, given the data and the model, of obtaining the tip character states if you start with that character state at the root. In other words, we keep track of the likelihood for the tipward parts of the tree, including our data, if the node we are considering had each of the possible character states. This calculation is:

$$L_P(i) = (\sum_{x \in k} Pr(x|i, t_L) L_L(x)) \cdot (\sum_{x \in k} Pr(x|i, t_R) L_R(x)) \tag{8.7.1}$$

where $i$ and $x$ are both indices for the $k$ character states, with sums taken across all possible states at the branch tips ($x$), and terms calculated for each possible state at the node ($i$). The two pieces of the equation are the left and right descendant of the node of interest. Branches can be assigned as left or right arbitrarily without affecting the final outcome, and the approach also works for polytomies (but the equation is slightly different). Furthermore, each of these two pieces itself has two parts: the probability of starting and ending with each state along the two branches being considered, and the current conditional likelihoods that enter the equation at the tips of the subtree ($L_L(x)$ and $L_R(x)$). Branch lengths are denoted as $t_L$ and $t_R$ for the left and right, respectively.

One can think of the likelihood "flowing" down the branches of the tree, and conditional likelihoods for the left and right branches get combined via multiplication at each node, generating the conditional likelihood for the parent node for each character state ($L_P(i)$).

Consider the subtree leading to species A and B in the example given. The two tip character states are 0 (for species A) and 1 (for species B). We can calculate the conditional likelihood for character state 0 at node 1 as:

$$L_P(0) = \left(\sum_{x \in k} Pr(x|0, t_L = 1.0) L_L(x)\right) \cdot \left(\sum_{x \in k} Pr(x|0, t_R = 1.0) L_R(x)\right) \tag{8.7.2}$$

Next, we can calculate the probability terms from the probability matrix **P**. In this case $t_L$ = $t_R$ = 1.0, so for both the left and right branch:

$$\mathbf{Q}t = \begin{bmatrix} -2 & 1 & 1 \\ 1 & -2 & 1 \\ 1 & 1 & -2 \end{bmatrix} \cdot 1.0 = \begin{bmatrix} -2 & 1 & 1 \\ 1 & -2 & 1 \\ 1 & 1 & -2 \end{bmatrix} \tag{8.7.3}$$

So that:

$$\mathbf{P} = e^{Qt} = \begin{bmatrix} 0.37 & 0.32 & 0.32 \\ 0.32 & 0.37 & 0.32 \\ 0.32 & 0.32 & 0.37 \end{bmatrix} \tag{8.7.4}$$

Now notice that, since the left character state is known to be zero, $L_L(0)$=1 and $L_L(1)$=$L_L(2)$=0. Similarly, the right state is one, so $L_R(1)$=1 and $L_R(0)$=$L_R(2)$=0.

We can now fill in the two parts of equation 8.2:

$$\sum_{x \in k} Pr(x|0, t_L = 1.0) L_L(x) = 0.37 \cdot 1 + 0.32 \cdot 0 + 0.32 \cdot 0 = 0.37 \tag{8.7.5}$$

and:

$$\sum_{x \in k} Pr(x|0, t_R = 1.0) L_R(x) = 0.37 \cdot 0 + 0.32 \cdot 1 + 0.32 \cdot 0 = 0.32 \tag{8.7.6}$$

So:

$L_P(0)$=0.37 · 0.32 = 0.12.

This means that under the model, if the state at node 1 were 0, we would have a likelihood of 0.12 for this small section of the tree. We can use a similar approach to find that:

(eq. 8.13)

$L_P(1)$=0.32 · 0.37 = 0.12.

$L_P(2)$=0.32 · 0.32 = 0.10.

Now we have the likelihood for all three possible ancestral states. These numbers can be entered into the appropriate boxes:

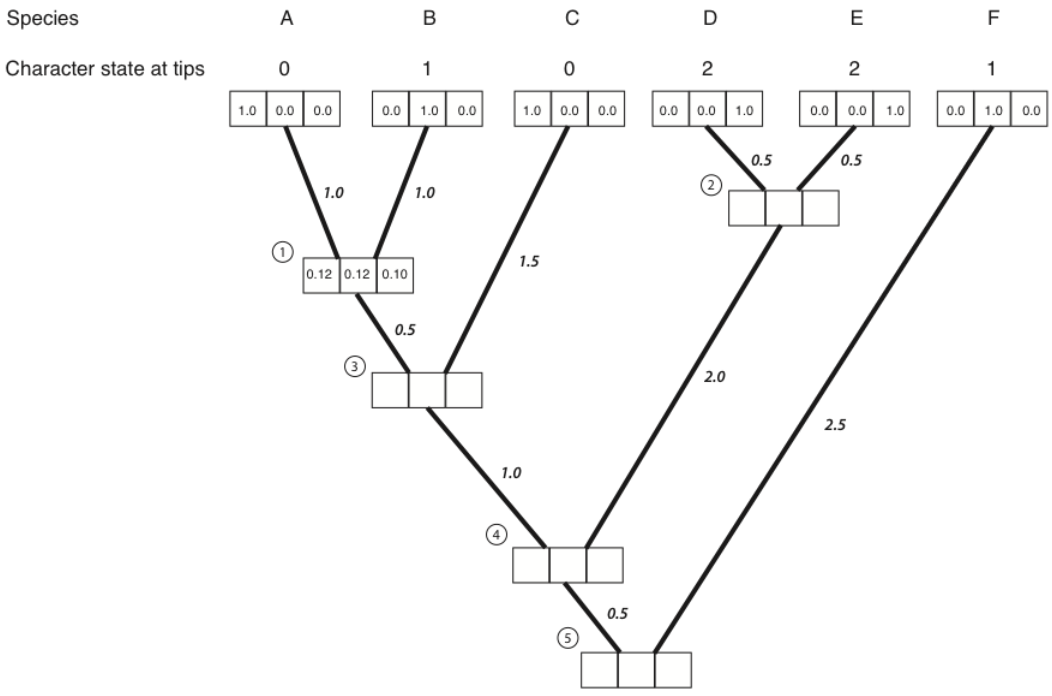


Figure 8.4C. Conditional likelihoods entered for node 1. Image by the author, can be reused under a CC-BY-4.0 license.

4. We then repeat the above calculation for every node in the tree. For nodes 3-5, not all of the $L_L(x)$ and $L_R(x)$ terms are zero; their values can be read out of the boxes on the tree. The result of all of these calculations:

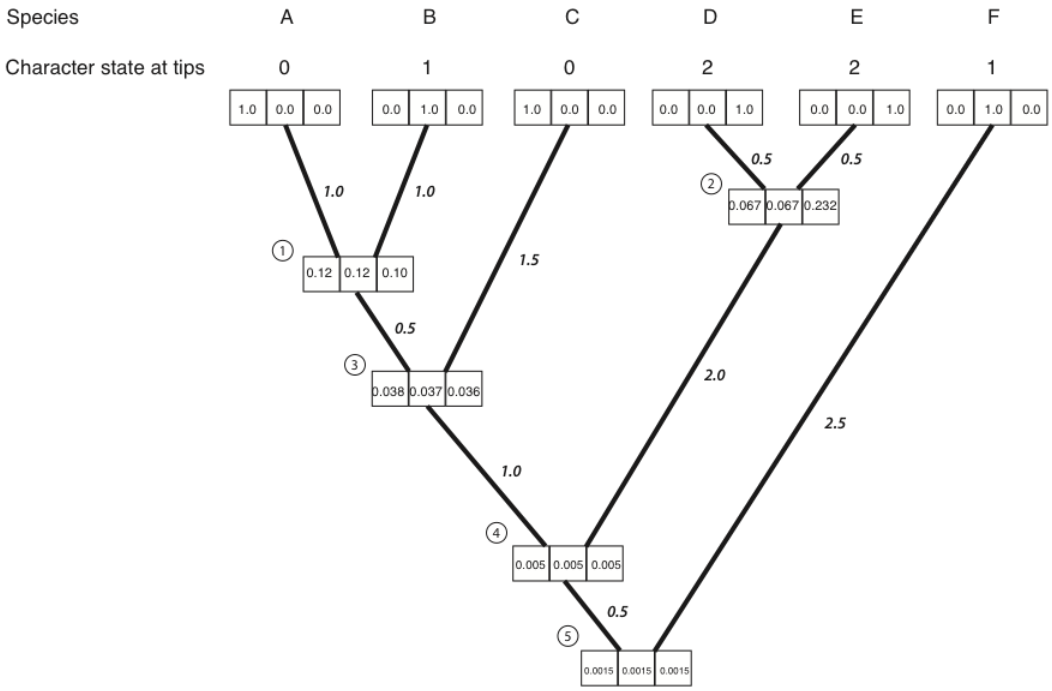


Figure 8.4D. Conditional likelihoods entered for all nodes. Image by the author, can be reused under a CC-BY-4.0 license.

5. We can now calculate the likelihood across the whole tree using the conditional likelihoods for the three states at the root of the tree.

$$L = \sum_{x \in k} \pi_x L_{root}(x) \tag{8.7.7}$$

Where $\pi_x$ is the prior probability of that character state at the root of the tree. For this example, we will take these prior probabilities to be uniform, equal for each state ($\pi_x$ = 1/$k$ = 1/3). The likelihood for our example, then, is:

$$L = 1/3 \cdot 0.00150 + 1/3 \cdot 0.00151 + 1/3 \cdot 0.00150 = 0.00150 \tag{8.7.8}$$

Note that if you try this example in another software package, like GEIGER or PAUP*, the software will calculate a ln-likelihood of -6.5, which is exactly the natural log of the value calculated here.

---